Energy-aware flow allocation algorithm for Energy Efficient Ethernet networks

I. Seoane*, J. A. Hernández*, P. Reviriego†, D. Larrabeiti*
*Department of Telematics Engineering
Universidad Carlos III de Madrid
Email: {iseoane, jahgutie, dlarra} @it.uc3m.es
†Computer Architecture and Technology Group
Universidad Antonio de Nebrija
Email: previrie@nebrija.es

***Abstract*—Energy Efficient Ethernet, as defined by the IEEE 802.3az standard, has shown not to be as efficient as originally expected given the large values of the transition times between the active and sleep power modes. In fact, EEE performs nearly optimal only when the link load is either very low or very high, but never at medium loads. So, in order to achieve large power savings, then it is necessary to design a flow allocation algorithm that allocates traffic demands on links that avoid medium traffic loads on links, since these are far from optimal.**

This work defines an EEE-FA, an energy-aware flow allocation algorithm that computes the best possible route in terms of energy consumption for a given network load condition. Essentially, EEE-FA computes the K-shortest paths for a given traffic demand and evaluates the consumption impact of allocating the traffic demand on each of them, in order to further select that route with minimum energy consumption impact for a given network status. This algorithm is compared with shortest path routing and it is shown that important energy savings may be achieved, however at the expense of increasing the global network traffic load and the average number of hops per demand as a consequence of using sub-optimal (in terms of distance) routes.

I. Introduction

Reducing the energy consumption of communications equipment is a research topic of growing interest with clear economical and environmental benefits. As noted in previous studies [1], the amount of energy consumed in IT is around 800 TWh per year, while a large portion of it may be saved just by designing more efficient hardware and software. Indeed, important energy savings may be achieved by dynamically selecting the most appropriate link speed for a communications link [2], [3] or by switching off unnecessary equipment [4], [5], [6].

In Ethernet, for instance, there is large room for energy improvement because: (1) Currently, Ethernet NICs consume 100% of its power even when idle; and (2) Ethernet is massively deployed. Indeed, Ethernet is present on most of corporative and residential local area networks, and moreover is beginning to expand to metropolitan area networks and maybe to backbone networks thanks to the latest advances in 40/100 G Ethernet. So any energy optimisation feature designed for Ethernet would translate into a great consumption reduction given its ubiquity.

In light of this, the IEEE [7] has recently approved the 802.3az Energy Efficient Ethernet standard which attempts to achieve large energy savings by defining two power modes: active and sleep. The idea is to put the Ethernet PHY into the sleep (low-power) mode when no data is pending for transmission. The sleep mode only consumes about 10% of the power spent in the active mode. Thus, low-loaded links are expected to benefit from the sleep mode since they are very likely to spend large periods of time in such a low-power mode.

Although this mechanism was originally expected to bring large power savings, previous studies from the authors have demonstrated that this is not actually the case given the large values of the mode-transition timers. In fact, EEE deviates very significantly from the proportional load vs consumption plot, which is often considered the ideal energy profile [8]. The reason for this suboptimal operation of EEE lies in the fact that the transition between the active and sleep modes is very slow and wastes a lot of energy. For instance, the transmission of a 1500-byte frame over a 1000 BASE-T Ethernet link has the following energy efficiency η:

$$\eta = \frac{T_{\text{frame}}}{T_w + T_{\text{frame}} + T_s} = \frac{1.2\mu s}{8.56\mu s} = 0.14 \qquad (1)$$

since $4.48\mu s$ and $2.88\mu s$ out of $8.56\mu s$ are spent in waking up and sleeping down the link respectively, and only $1.2\mu s$ on actual data transmission. That is, 86% of the power is spent only on switching the PHY between the two power modes. This makes the power consumption versus traffic load plot look like Fig. 1 for 1000 BASE-T [8].

As shown, *the energy consumption of EEE links is nearly 100% for traffic loads greater than 10% for 1000 BASE-T, 40% for 10GBASE-T and 50% for 100 BASE-TX*. Nevertheless, some constructive conclusions can be identified from Fig. 1: There are two ranges of traffic load at which the difference between the EEE consumption-load plot and the optimal proportionality line is small: at very low traffic loads (0-5%) and at very high traffic loads (60% onwards).

In light of this, the goal of this work is to define an energy-aware flow allocation algorithm (in what follows *EEE-FA*) that computes the best possible path in terms of traffic consumption for all traffic demand arrivals. The algorithm

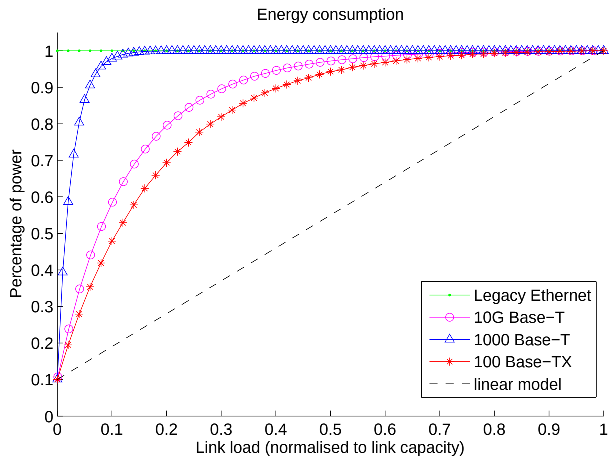


Fig. 1. Energy consumption versus traffic load for 100BASE-TX, 1000BASE-T and 10GBASE-T EEE, as shown in [8]

computes the K-shortest paths for an incoming given traffic demand and evaluates the consumption impact of allocating such a demand on each path, taking into account the traffic demands already allocated on the network. Then, the algorithm selects the path with minimum energy consumption (impact) on the network. This algorithm is compared with shortest path routing (i.e. $K = 1$-shortest path) and it is shown that important energy savings may be achieved, however at the expense of increasing the average number of hops per demand and the global network load as a consequence of using sub-optimal (in terms of distance) paths.

A similar idea was proposed in [9], however with several important differences:

- First, our algorithm computes the best possible path in terms of energy impact on a per-demand arrival basis. The work in [9] formulates the flow allocation optimisation problem, where an energy-consumption objective function needs to be minimised subject to a set of constraints. Such a centralised algorithm is solved for a given traffic matrix that specifies all traffic demands, therefore, these need to be known beforehand, which might not be suitable in certain cases.
- Secondly, our algorithm uses the real 100BASE-TX, 1000BASE-T and 10GBASE-T power consumption vs load curves as obtained after applying EEE to Poisson traffic. The work in [9] uses a set of theoretical power consumption vs load curves not really related with EEE.

The reminder of this work is organised as follows: Section II presents the EEE-FA algorithm and demonstrates with a three-node example its operation and potential energy savings. Section III performs a deep simulation study of the EEE-FA on the NSFNet topology. Finally, Section IV summarises the main findings and contributions of this work.

II. The Energy Efficient Ethernet Flow Allocation example (EEE-FA)

A. A three-node network EEE-FA example

Let us consider the three-node (triangle) network shown in Fig. 2 for an example on how to apply EEE to the flow allocation decisions. The three links connecting nodes N_1, N_2 and N_3 are all of capacity C. Additionally, let us assume three traffic demand arrivals only, which must be mapped on the triangle network:

- First, a demand $d_{13}^{(1)} = 0.5C$ from node 1 to node 3 of 50% of the link capacity.
- A second demand $d_{32}^{(2)} = 0.5C$, from node 3 to node 2, also of 50% of the link capacity.
- And finally, a third demand arrival $d_{12}^{(3)} = 0.2C$, from node 1 to node 2, of 20% of the link capacity this time.

Shortest-path routing would allocate the three traffic demands as shown Fig. 2 (dashed lines), making use of the three direct links. In such a case, the average load per link would be:

$$E[\rho_l] = \frac{0.5C + 0.5C + 0.2C}{3} = 0.4C$$

while the average energy consumption per link (normalised) asuming that the network uses a 100BASE-TX infrastructure (Fig. 1):

$$E[P] = \frac{0.95 + 0.95 + 0.7}{3} = 0.86$$

and average number of hops per demand:

$$E[H] = \frac{1 + 1 + 1}{3} = 1$$

Alternatively, the third traffic demand might be mapped on the two links already active, as shown in Fig. 2(b), thus increasing the average load and number of hops to

$$E[\rho_l] = \frac{0.7C + 0.7C + 0}{3} = 0.47C$$

and

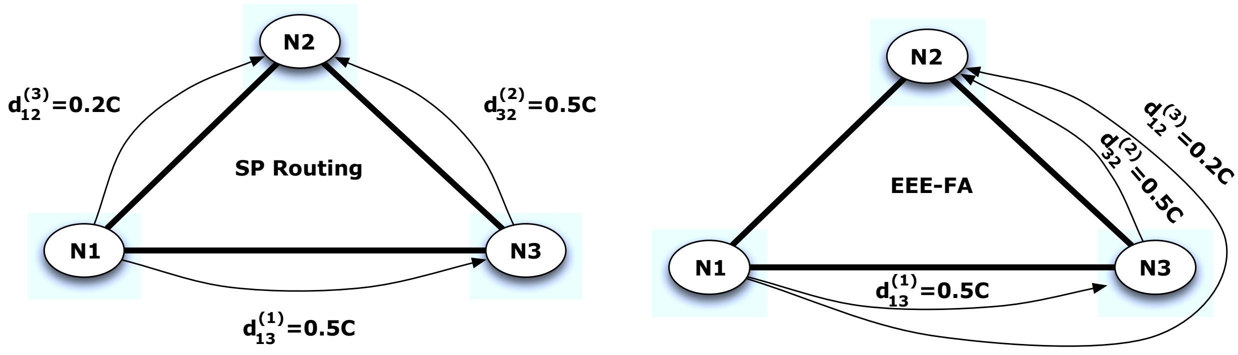
$$E[H] = \frac{1 + 1 + 2}{3} = 1.33$$

respectively, but reducing the total energy consumption to:

$$E[P] = \frac{0.98 + 0.98 + 0.1}{3} = 0.69$$

Clearly, allocating the third traffic demand over two already active links saves energy mainly because this strategy permits to have one link with null load (hence 10% of power consumption only). Additionally, the two links already active suffer little energy increase for allocating the third demand (from 0.95 at $0.5C$ load to 0.98 at $0.7C$ load) as noted from the power vs load figure (Fig. 1).

Next section defines the EEE-FA algorithm which attempts to reduce the average power consumption on a network following the same premises of this example.



(a) SPR allocation

(b) EEE-FA

Fig. 2. Example of routing three demands in a 3-nodes network using different cost approach: (a) Shortest-Path Routing (SP-Routing), (b) Energy Efficient Ethernet Routing (EE-Routing)

B. The EEE-FA algorithm

The EEE-FA algorithm, summarised in Table I uses the following notation:

- N the set of nodes in the network.
- L the set of links in the network.
- D the total number of traffic demands to be allocated.
- $d_i(m,n)$ the i-th traffic demand with source node $m \in N$ and destination node $n \in N$.
- p_{ik} the sorted list of links traversed by the k-th shortest path of demand $d_i(m,n)$ (remark that $k = 1,\ldots,K$).

Now, for every demand $d_i(m,n)$, with $i = 1,\ldots,D$, the EEE-FA first computes the K-shortest paths p_{ik}, $k = 1,\ldots,K$ between nodes m and n. For every, possible path p_{ik}, then EEE-FA evaluates the energy cost of allocating d_i on every path p_{ik} as:

$$C_{ik} = \sum_{l \in p_{ik}} \left[f_j\left(\rho(l) + d_i(m,n)\right) - f_j\left(\rho(l)\right) \right]$$

which basically comprises the energy consumption after allocating d_i on the links specified by the k-th shortest path p_{ik}. Here, the $f_j(.)$ refer to the load vs cost functions shown in Fig. 1, for 100BASE-TX, 1000BASE-T and 10GBASE-T. That is, $f_{100BASE-TX}(\rho(l))$ gives the energy consumption of link $l \in L$ at load $\rho(l)$.

Finally, EEE-FA selects the winner path p_i^* that minimises the energy impact C_i^* for demand $d_i(m,n)$:

$$\text{Select } p^* \text{ that minimises } C_i = \min_k C_{ik}$$

In summary, the EEE-FA computes the K-shortest paths for incoming traffic demands. After that, computes the energy impact of allocating incoming traffic demand on every path from the K-shortest ones, and finally chooses that path with minimum energy impact on the network given its present traffic load. The EEE-FA is summarised in Table I.

III. Experiments

This section shows the potential of the EEE-FA algorithm in terms of energy savings. The EEE-FA algorithm is compared against shortest path routing in terms of average energy consumption per link, average link load and average number of hops per traffic demand. It is worth noticing that the shortest path allocation of traffic demands (in what follows SPR) is the same as EEE-FA with parameter $K = 1$, that is, the shortest path to destination is always selected and its energy impact evaluated.

A. Simulation scenario

In this experiment, we have used the NSFNet network topology of Fig. 3 whose main parameters are summarised in Table. II. Remark that the number of possible source-destination pairs is $N(N-1)$, and all links are assumed bidirectional.

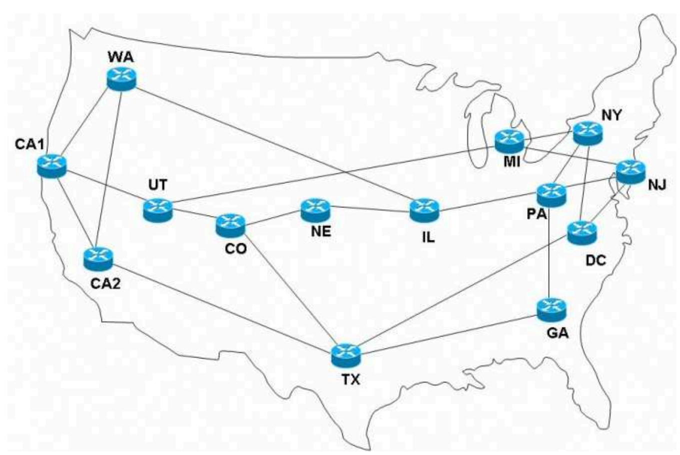


Fig. 3. The NSFnet topology

As noted in Table II, traffic demands are assumed to arrive at the network following a Poisson process with rate λ demands per unit of time, require $h_D = 0.1C$ fixed units of capacity per demand and are allocated on a given path for an exponentially distributed amount of time with mean $1/\mu$ units of time. The

1	For every new incoming demand $d_i(m,n)$
2	Compute the K-shortest paths whose source is m and destination n.
3	For every path p_{ik}, with $k=1,\ldots,K$ of demand d_i
4	Compute the energy impact $C_{ik} = \sum_{l \in p_{ik}} \left[f_j\left(\rho(l) + d_i(m,n)\right) - f_j\left(\rho(l)\right)\right]$ for every path p_{ik}
5	Choose p^* that minimises $C_i = \min_k C_{ik}$
6	Allocate demand d_i through path p^* if there is enough capacity for it. Otherwise, repeat step 3 without that path
7	Update the new value of $\rho(l)$ for each link of the path p^*

TABLE I
THE EEE-FA ALGORITHM

Parameter	NSFnet
Number of nodes N	14
Number of bidirectional links L	40
Average Network Load	1%, 5%,10%
	25%, 50% 70%
Capacity requested per traffic demand h_D	$0.1C$ fixed
Average connection size $E(S)$	100 MBytes

TABLE II
NSF NETWORK SIMULATION PARAMETERS

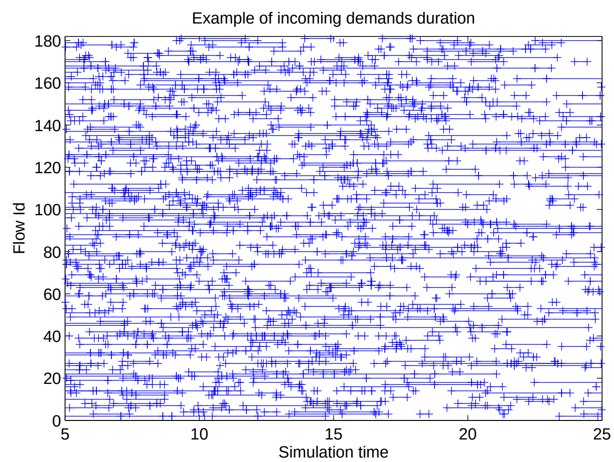


Fig. 4. An example of simulated traffic demands on the NSFnet

source and destination of each demand nodes are randomly drawn from the total set of nodes in the network, with the only restriction that the same node cannot be source and destination for any demand. This yields a number of $14 * 13 = 182$ possible source-destination pairs. Each simulation considers 1820 end-to-end demands, which is about 10 demands or flows per source-destination pair.

Now, the average flow duration $1/\mu$ is computed assuming an average connection size of $E(S) = 100 Mbytes$ bytes. In other words:

$$\frac{1}{\mu} = \frac{8 \times E(S)}{h_D}$$

which is $8s$ for 100BASE-TX, $0.8s$ for 1000BASE-T and $0.08s$ for 10GBASE-T.

Finally, with that value of μ, the traffic demand rate λ is obtained as:

$$\lambda = \rho\mu$$

where ye$\rho = 0.05$ (low loads), 0.2 (medium loads) or 0.7 (high loads).

For instance, Fig.4 shows an simulation scenario of the total end-to-end demands at medium traffic load. In this case, the simulation lasted for about 25 units of time, but such duration varies depending on the network load and the simulation scenario.

B. Experimental results

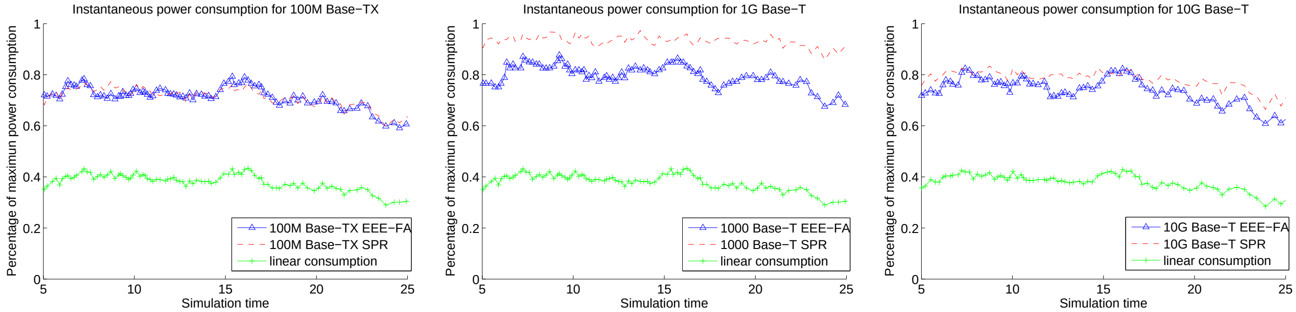
Fig.5 shows the instantaneous power consumption for one simulation scenario (that means, same set of demands) for 100BASE-TX, 1000BASE-T and 10GBASE-T, at medium traffic loads ($\rho = 0.25$, medium loads). As shown, very large energy savings may be achieved for 1000BASE-T, but smaller for 100BASE-TX and 10GBASE-T. Essentially, 1000BASE-T offers more room for improvement because it has the highest transition times between the active and idle power modes, which results in the worst consumption versus load plot (Fig. 1).

Finally, Table III shows a summary of results obtained for about 20 simulations of 1820 end-to-end traffic demands with 18 different experimental setups, ranging:

- Algorithm: SPR and EEE-FA.
- Link capacity: 100BASE-TX, 1000BASE-T and 10GBASE-T (100M, 1G and 10G in the table).
- Link load: low ($\rho = 0.01$ and $\rho = 0.05$), medium ($\rho = 0.1$ and $\rho = 0.25$) and high ($\rho = 0.5$ and $\rho = 0.7$).

For each simulation, we compute the avg. number of hops $E(H)$, avg. link load $E(\rho_l)$ and avg. power consumption $E(P)$ over time and then, we finally averaged for each experiment setup. The results are summarised in Table III.

As shown in the table, the EEE-FA algorithm yields important energy savings at medium traffic loads (25%) and moderate energy savings at both low and high traffic loads. Additionally, such energy savings are obtained at the expense of increasing the average number of hops per traffic demand, which clearly results in a global performance degradation due to the increased network load. Finally, the percentage of blocked demands, that is, those which cannot be allocated on any path is larger when using EEE-FA than with SPR because



(a) 100BASE-TX (b) 1000BASE-T (c) 10GBASE-T

Fig. 5. Instant power consumption for EEE-FA and SPR at different link capacities

(a) Low loads

		$\rho = 0.01$			$\rho = 0.05$		
		100M	1G	10G	100M	1G	10G
Avg. number of hops	SPR	2.20	2.20	2.20	2.20	2.20	2.20
	EEE-FA	2.21	2.30	2.25	2.21	2.30	2.25
Avg. link load	SPR	1.093%	1.093%	1.093%	1.115%	1.115%	1.115%
	EEE-FA	1.103%	1.151%	1.124%	1.124%	1.173%	1.145%
Avg. power consumption	SPR	12.66%	17.28%	14.20%	12.76%	17.46%	14.32%
	EEE-FA	12.64%	17.13%	14.19%	12.74%	17.31%	14.30%
Blocked demands	SPR	0	0	0	0	0	0
	EEE-FA	0	0	0	0	0	0

(b) Medium Loads

		$\rho = 0.10$			$\rho = 0.25$		
		100M	1G	10G	100M	1G	10G
Avg. number of hops	SPR	2.20	2.20	2.20	2.20	2.20	2.20
	EEE-FA	2.21	2.30	2.25	3.35	3.31	3.48
Avg. link load	SPR	1.123%	1.123%	1.123%	34.67%	34.67%	34.67%
	EEE-FA	1.132%	1.183%	1.153%	53.76%	52.71%	54.96%
Avg. power consumption	SPR	12.76%	17.50%	14.32%	61.11%	81.29%	67.32%
	EEE-FA	12.64%	17.35%	14.30%	61.06%	68.27%	63.67%
Blocked demands	SPR	0	0	0	0	0	0
	EEE-FA	0	0	0	0.073%	0.037%	0.055%

(c) High Loads

		$\rho = 0.50$			$\rho = 0.70$		
		100M	1G	10G	100M	1G	10G
Avg. number of hops	SPR	2.22	2.22	2.22	2.35	2.35	2.35
	EEE-FA	3.20	2.98	3.27	2.97	2.74	3.02
Avg. link load	SPR	56.22%	56.22%	56.22%	56.69%	55.80%	56.61%
	EEE-FA	82.42%	73.18%	82.41%	66.85%	61.73%	67.33%
Avg. power consumption	SPR	73.67%	85.79%	77.97%	80.68%	85.03%	82.13%
	EEE-FA	76.56%	78.69%	78.58%	81.39%	82.88%	82.40%
Blocked demands	SPR	0	0	0	1.172%	1.172%	1.172%
	EEE-FA	2.454%	0.512%	2.931%	8.882%	3.864%	8.260%

TABLE III
SUMMARY OF SIMULATION RESULTS

the average traffic load is much higher, thus resulting in more links full.

In conclusion, the EEE-FA must be used on scenarios that admit large energy savings while, at the same time, the performance degradation of choosing sub-optimal (long paths) does not increase the global network load excessively.

IV. Summary and conclusions

This work proposes EEE-FA, a novel flow allocating algorithm that attempts to improve the energy effiency of an Ethernet network in terms of energy consumption as a function of its links load. This algorithm takes into account the energy increment of allocating incoming flows on different paths, and selects that one with minimum energy impact. This is done on a per-demand arrival basis, which facilitates its implementation on networks with random flow arrivals.

The experimental results show that important energy savings may be achieved with EEE-FA at certain scenarios, mainly at medium traffic loads and for 1000BASE-T. However, such energy reduction is achieved at the expense of degrading other aspects of network performance and administration: longer end-to-end paths and higher traffic loads. This will definetily translate into large end-to-end delays experienced by the traffic flows.

Acknowledgements

The authors would like to thank the support of the T2C2 Spanish project under code TIN2008-06739-C04-01, and the CAM-UC3M Greencom research grant under code CCG10-UC3M/TIC-5624.

References

[1] K. Christensen, “Green networks: Reducing the energy comsumption of networks,” http://www.csee.usf.edu/ christen/energy/madridTalk10.pdf.

[2] C. Gunaratne, K. Christensen, S. Suen, and B. Nordman, “Reducing the energy consumption of ethernet with an adaptive link rate (alr),” *IEEE Transactions on Computers*, vol. 57, no. 4, pp. 448–4561, April 2008.

[3] P. Reviriego, B. Huiszoon, V. López, R. B. Coenen, J. A. Hernández, and J. A. Maestro, *Accepted for publication in IEEE J. Selected Topics in Quantum Electronics*.

[4] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, “Power awareness in network design and routing,” in *In Proc. IEEE INFOCOM*, 2008.

[5] E. Gelenbe and S. Silvestri, “Reducing power consumption in wired networks.” in *ISCIS*. IEEE, 2009, pp. 292–297. [Online]. Available: http://dblp.uni-trier.de/db/conf/iscis/iscis2009.html#GelenbeS09

[6] ——, “Optimisation of power consumption in wired packet networks,” in *QSHINE*, 2009, pp. 717–729.

[7] I. 802.3az Energy Efficient Ethernet Task Force, “Ieee 802.3az energy efficient ethernet standard,” Draft Amendment to IEEE Std 802.3-2008, 2010.

[8] P. Reviriego, J. A. Hernández, D. Larrabeiti, and J. A. Maestro, “Performance evaluation of energy efficient ethernet,” *IEEE Communications Letters*, vol. 13, no. 9, pp. 697–699, Sept. 2009.

[9] J. C. C. Restrepo, C. Gruber, and C. M. Machuca, “Energy profile aware routing,” in *In First International Workshop on Green Communications IEEE International Conference on Communications (ICC)*, June 2009; Dresden, Germany.